[**Algoritmi**](#_kvn4k77ib3tz) **3**

[Algoritmo Distribuzione Uniforme](#_yep4vnctcokx)

[INGLESE](#_xat8oprmlv67)

[GIAPPONESE](#_zdup5wbj2cmq)

[Algoritmo Distribuzione Non Uniforme](#_nb70w3oh08cj) 3

[Esempio](#_pi6vx2prcvx) 3

[INGLESE](#_tdbq5y7rc9qp)

[GIAPPONESE](#_mmz9dxwyyeho)

[Generatore Lineare Congruenziale](#_vqb2m94f8klr) 4

[INGLESE](#_qu70uvpuch5b)

[GIAPPONESE](#_n8xeceng30a2)

[Generatore di Fibonacci ritardato](#_by5v163qdjt4) 5

[INGLESE](#_qvmuz0nhxsvs)

[GIAPPONESE](#_n977ce4coho3)

[Registro a scorrimento a Retroazione Lineare](#_uvnufavnctj) 6

[INGLESE](#_j8g5452766nb)

[GIAPPONESE](#_3sl0tnti95hf)

[Blum Blum Shub](#_o94fvvhlyjw) 7

[INGLESE](#_k8b8oop9qles)

[GIAPPONESE](#_isuv8229lo06)

[**Frattali**](#_lfrpbl2qhbgq) **9**

[Triangolo di Serpinsky](#_qwjsryrd5n9m) 9

[INGLESE](#_3z90y07nk31y)

[GIAPPONESE](#_ici53d9u5m79)

[T-Square](#_mkigrri4rr1n) 9

[INGLESE](#_tll1x68u1sba)

[GIAPPONESE](#_q44cr6srn01d)

[Shuriken](#_fwlaxu92bio1) 10

[INGLESE](#_tv104ttasicp)

[GIAPPONESE](#_eyyix2n6oun0)

[Forest](#_uk6arqoscvio) 10

[INGLESE](#_9p2cvovzdtsv)

[GIAPPONESE](#_6w1iexorgkcr)

[Quad](#_funw0ytjcus5) 11

[INGLESE](#_qby39fkicjz8)

[GIAPPONESE](#_wroq6y8j1bga)

[Pent Serp](#_2ekdnbimjhtm) 12

[INGLESE](#_z521xo4j77tv)

[GIAPPONESE](#_6ckgms273k9t)

[Hive](#_d8cqzhchm29y) 12

[INGLESE](#_kop54ktp9bnd)

[GIAPPONESE](#_a32gbz5a3rk8)

[Star](#_4j9t78m4pc2j) 13

[INGLESE](#_ytcx544hhi62)

[GIAPPONESE](#_o4efjp5ind3f)

[Felce di Barnsley](#_iv923e75jug) 13

[INGLESE](#_ojcgs7ba38fn)

[GIAPPONESE](#_l30mgy31m7i)

["xの範囲は \n" + //TODO](#_osj2odnvuo9y) 14

["xの範囲は \n" + //TODO](#_yylogba32aj) 14

[Set di Mandelbrot](#_f5aqxhiq2w0a) 14

[INGLESE](#_qlv8kvu0cp25)

[" "; //TODO but how?](#_mdxcyd1kujra) 14

[GIAPPONESE](#_zgdz8571kohe)

["xの式は " + //TODO](#_zhr590yisvsw) 14

["" + //TODO IF THEN COLORI](#_34u60etp9rs7) 14

# 

# **Algoritmi**

## Algoritmo Distribuzione Uniforme

#### INGLESE

string ris = "There are different types of distributed generation algorithms for pseudo-random numbers.\n " +

"The ones implemented in this application are:\n" +

"\Linear congruential generator;\n" +

"\tLagged Fibonacci generator;\n" +

"\tLinear-feedback shift register;\n" +

"\nBlum Blum Shub;\t" +

"";

#### GIAPPONESE

string ris = "今日の世界にたくさん分散配置擬似乱数生成器があります.\n " +

"このアップに四があります:\n" +

"\t合目的的線状擬似乱数生成器;\n" +

"\t遅れフィボナッチの探索擬似乱数生成器\n" +

"\t線状視野移動遡及的登記簿\n" +

"\nブラムブラムシャブ;\t" +

"";

## Algoritmo Distribuzione Non Uniforme

### Esempio

#### INGLESE

return "This is an example of a non-uniform distributed algorithm .\n"+

"The initial function is "+Nome+";\n"+

"The primitive is "+PrimitivaScelta+";\n"+

"The inverted function of the primitive is "+FunzioneInversa+";\n" +

"The final function explicated in x is "+FunzioneFinale+";\n"+

"Taking as input a decimal number generated by GLC,\nthe less significant digits are taken into consideration to get the final number" ;

#### GIAPPONESE

return "これは偏在の擬似乱数生成器の例\n"+

"初めの式は "+Nome+";\n"+

"インテグラルは "+PrimitivaScelta+";\n"+

"インテグラルのf,xの逆関数は"+FunzioneInversa+";\n" +

"終わりのxの式は "+FunzioneFinale+";\n"+

"インプットは合目的的線状擬似乱数生成器の十進数アウトプットです。\n終わりのアウトプットは十進数アウトプットの最下位値" ;

### Generatore Lineare Congruenziale

#### INGLESE

string ris = "The linear congruential generator is an old algorithm, that is easy to reproduce and computationally light.\n " +

"Its recursive formula is: X(n+1) = [ a \* X(n) + c] mod m\n" +

"Where:\n" +

"\tm it's the module between 0 and +infinite (in this case m = "+ M+"); \n" +

"\ta it's the multiplier between 0 and m (in this case a = "+A+");\n" +

"\tc it's the increment between 0 and m (in this case c = "+C+");\n" +

"\tX(0) it's the seed between 0 and m (in this case X(0) = "+Xi+");\n" +

"The period of the algorithm is m, and it's full only if:\n" +

"\tc and m are coprime (they have as MCD 1);\n" +

"\ta-1 is divisible for all of m's prime factors;\n" +

"The linear congruential generator was used by GCC with the rand function and Java's Random class.\n" +

"It's not efficient and it shouldn't be used where a high degree of casualty is required or for cryptography.\n";

#### GIAPPONESE

return "合目的的線状擬似乱数生成器は古くて、易しいです.\n " +

"再帰の式は : X(n+1) = [ a \* X(n) + c] mod m\n" +

"\tm は加群、m大なり0、m小なり無限大(今のmは "+ M+"です); \n" +

"\ta は追補 、a大なり0、a小なりm(今のaは "+ A+"です); \n" +

"\tc は乗数、c大なり0、c小なりm(今のcは "+ C+"です); \n" +

"\tX0 は種、X0大なり0、X0小なり無限大(今のX0は "+ Seed+"です); \n" +

"演算手順の周期はm。\n" +

"合目的的線状擬似乱数生成器は GCC の rand とJavaの Random class にあります .\n" +

"この演算手順を暗号化手法に用いないでください！\n";

### Generatore di Fibonacci ritardato

#### INGLESE

return "The lagged Fibonacci generator is an algorithm for pseudo-random number generation based on a generalisation of the Fibonacci sequence.\n" +

"From the definition of the Fibonacci sequence:\n" +

"the generator is defined as F(n) = [ F(n-j) @ F(n-k) ] mod m, with 0 < j < k <= n \n" +

"Where:\n" +

"\tF(n) is the last term of the succession\n" +

"\tF(n-j) and F(n-k) are any of the two previous terms of the succession\n" +

"\t@ is a binary operator(+,-,\*,/,XOR,AND,ecc.)\n" +

"In this case, it uses:\n" +

"\tj = "+j+"\n" +

"\tk = "+k+"\n" +

"\tn = "+n+"\n" +

"Properties:\n" +

"\t- As all pseudo-random number generators, the lagged Fibonacci generator is a periodic function.\n" +

"\t- The maximum period varies depending on the operator used. In case of a sum or subtraction, the generator has the period equal to p such that\n" +

"\t 2^{k-1} \* 2^{m-1}\n" +

"\t If it's a multiplication instead\n" +

"\t 2^{k-1} \* 2^{m-3}\n" +

"\t The period of the multiplication is a quarter of the one of the sum.\n";

#### GIAPPONESE

return "これは遅れフィボナッチの探索の擬似乱数生成器.\n" +

"擬似乱数生成器の式は F(n) = [ F(n-j) @ F(n-k) ] mod m,、j大なり0、j大なりk、n大なりk\n" +

"\tF(n) は最後の値\n" +

"\tF(n-j) と F(n-k)はこの遅れフィボナッチの探索の上値\n" +

"\t@ は二項演算子(+,-,\*,/,XOR,AND,ecc.)\n" +

"今のjは "+ j+"です\n"+

"今のkは "+ k+"です\n"+

"今のnは "+ n+"です\n"+

"遅れフィボナッチの探索は定期的式です\n"+

"＋の周期の範囲は 2^{k-1} \* 2^{m-1}です\n"+

"×の周期の範囲は 2^{k-1} \* 2^{m-3}です\n"+

"×の周期の範囲は「＋の周期の範囲」分の4です";

### Registro a scorrimento a Retroazione Lineare

#### INGLESE

return "The linear-feedback shift register is a type of shift register where the input data is produced from a linear function of the internal state.\n" +

"The only linear function in bit are XOR and XNOR ;\n" +

"meaning that it's shift register where the input bits are produced by the xor of some of the memorized bits within the registers.\n" +

"The bit's position list that influcences the next sequence is called sequence tap.\n" +

"The outputs that influence the inputs are called tap.\n" +

"A maximal LFSR produces an n-sequence (aka. it passes through all of the possible shift registers except the one that produces all zeros),\n" +

"unless the initial state is all zeros, in that case the exit state remains the same.\n" +

"The tap sequence of a LFSR can be represented as a module 2 polynomial. This means that the polinom's coefficients must be either 1 or 0.\n" +

"This is known as a retroactive polynom.\n" +

"Properties:\n" +

"\t- If (and only if) this is a primitive polynom, then the LFSR is maximal\n" +

"\t- The LFSR is maximal only if the tap count is even\n" +

"\t- The tap values of a maximal LFSR are coprime\n" +

"\t- There can be more than one tap sequence that makes maximal a fixed length LFSR \n" +

"\t- Once found the maximal tap sequence, another one can be fount using an automatic procedure: if the sequence,\n" +

"\t in a n bit LFS, is [n,A,B,C], it's \"specular\" sequence is [n,n-A,n-B,n-C] (for example the sequence [32,3,2,1] has as its counterpart [32,31,30,29]).\n" +

"\t Both produce a maximal LFSR." +

"Properties of the exit sequence:\n" +

"\t- Ones and zeros follow one another (runs).\n" +

"\t The exit sequence 0110100, for example, is made of five runs, each having length 1,2,1,1,2.\n" +

"\t In a maximal LFSR period, there are 2^{n-1} runs (e.g., a 6 bit LFSR has 32 runs);\n" +

"\t and exactly 1/2 of those runs have one bit, 1/4 have 2 bit, untli the last run of zero n-1 bits,\n" +

"\t and one run of une n bits.\n" +

"\t- LFSR's exit sequences are deterministic: if the current state is known, the next one can be forecasted. \n" +

"LFSR's can be implemented in hardware, and that makes them useful in applications that reqiure a fast pseudo-rundom number generation,\n" +

"as in the r Direct Sequence Spread Spectrum radio technique, used in theUMTS.\n"+s

"The GPS used LFSR to quicly send a sequence equal to high-precision relative istances,\n" +

"taking advantage of it's determinism: that's because you only need to transmit the used seed to the transmitter since the generated sequence will be the same on the receiver .\n";

#### GIAPPONESE

return "線状視野移動遡及的登記簿のインプットはインターナルステータスのアウトプットです.\n" +

"式は XOR と XNOR だけです;\n" +

"インプットのビットは レギスターのビットです。\n" +

"意義深いビットは「タップ」です。\n" +

"LFSRに一番大きなの一連はN "+

"タップの列は二グレードポリノムですから,ポリノムのコエフィチアンツはゼロです,一です\n" +

" これは言うところの遡及的ポリノムです \n" +

"ハードウェアのイムプレゼントすることができますから、迚速い擬似乱数生成器にイムプレゼントすることは便利です\n" +

"\t例:レジオの術,直接拡散方式(DSSS)\n"+

"GPSも線状視野移動遡及的登記簿を用います" ;

### Blum Blum Shub

#### INGLESE

return "The algorithm uses four variables: q, p, n and a seed.\n" +

"\tp and q are two prime numbers,\n\tn è it's their product, \n\tthe seed is a coprime number lower than n.\n" +

"the starting value is set by the module of the seed two squared (X0 = seed ^ 2 mod n),\n" +

"to find the next X values simply apply the previous formula to the current variable (X(n) = X(n-1)^2 mod n\n)" +

"for each X(n) the less significant bit is taken and put into a bit vector.\n" +

"The result is achieved by the conversion of the binary number found in the array\n" +

"Eg: with n=5 the resulting binary number is = 10001; equal to the result = 17\n";

#### GIAPPONESE

return "演算手順のインプットは q, p, n ,seed.\n"+

"\tpとqは素数,\n\tnはpかけるq,\n\tseedは nの互いに素,seed小なりn\n"+

"X0は seed の 2 乗 mod n です,\n"+

"X(n) は X(n-1)の 2 乗 mod nです\n)"+

"そして,X(n)の最小有効ビットをビットの同位列にインサート\n"+

"アウトプットはビットの同位列の整数値です\n"+

"例\n\tnは5です,ビットの同位列は10001,アウトプットは17";

# 

# Frattali

## Triangolo di Serpinsky

### INGLESE

ris = "The Serpinsky’s triangle undoubtedly is one of the most famous fractals, mainly because it's the easiest one to create\n" +

"To create it via chaos game you need to proceed in the following way:\n\t" +

"1) Pick one of the triangle vertex as the starting point;\n\t" +

"2) The next point will be chosen at half of the distance between the current point and one of the vertex(randomly chosen)\n " +

"If you want to recreate it in the non-procedural way, you have to inscribe a triangle inside the initial one.\nThis triangle has the midpoints of the initial triangle sides as vertices .\n You must repeat the procedure with the other triangles, accordingly to the recursion level chosen.\n" +

"An infinitely iterated Serpinsky triangle has an area of 0.\n";

### GIAPPONESE

ris = "サーピンスキの三角形は世界で一番有名な自己相似図形です。創造は易しいですから。\n" +

"　混乱遊戯のサーピンスキの使い方は：\n\t" +

"一つ) 三角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"非ランダムサーピンスキの使い方は：\n" +

"一つ) さんかっけいの側の半ばは起点；\n" +

"サーピンスキの三角形の幾何学的領域はゼロ。\n";

## T-Square

### INGLESE

ris = "The T-Square is another famous fractal that unlike Serpinsky's triangle has one condition to follow for its creation: the next chosen vertex must not be the opposite of the current chosen one.\n" +

"To create it via chaos game you need to proceed in the following way:\n\t" +

"1) Pick one of the square vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point' \n" +

"If you want to recreate it in the non-procedural way you need to:\n\t " +

"1) Pick a starting square;\n\t" +

"2) Add a smaller square that has a starting triangle's vertex in the centre and is half the length of the starting triangle." +

"3) Repeat the procedure (and remember to follow the condition for its creation) until you have obtained the recursion level desired;\n\t";

### GIAPPONESE

ris = "Tの四角もとても有名です。\n" +

"Tの四角の使い方は：\n\t" +

"一つ) 四角形の頭頂は起点；\n\t" +

"二つ) 次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点は向い点ことができません。\n";

## Shuriken

### INGLESE

ris = "This shuriken-like nameless fractal can only be obtain via chaos game.\n" +

"To create it you need to proceed in the following way:\n\t" +

"1) Pick one of the square vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point'\n " +

"3)Condition: the next chosen vertex must not be counter-clockwise of the current chosen one\n "

### GIAPPONESE

ris = "この手裏剣的自己相似図形は混乱遊戯丈作られることができます。\n" +

"この自己相似図形の使い方は：\n\t" +

"一つ) 四角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点は反時計回り点ことができません。\n" +

## Forest

### INGLESE

ris = "This fractal can be obtained only via chaos game.\n" +

"To create it you need to proceed in the following way:\n\t" +

"1) Pick one of the square vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point'\n " +

"Condition) The next point's value can't have a difference of one from the current point's value\n" +

"The values are:\n" +

"1\t4\n2\t3\n"+

"The following example has been created with 100 000 points\n";

### GIAPPONESE

ris = "この森的自己相似図形は混乱遊戯丈作られることができます。\n" +

"この自己相似図形の使い方は：\n\t" +

"一つ) 四角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点の値はーことができません。\n" +

"値は:\n" +

"ー\t四\n三\t二\n" +

"この自己相似図形は十万点。";

## Quad

### INGLESE

ris = "This fractal can only be obtained via chaos game.\n" +

"1)the starting shape is a square\n"+

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point'\n " +

"Condition: the chosen point cannot be the same\n";

### GIAPPONESE

ris = "この自己相似図形は混乱遊戯丈作られることができます。\n" +

"使い方は：\n\t" +

"一つ) 四角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点は同じ点ことができません。\n";

## Pent Serp

### INGLESE

ris = "This fractal can be obtained only via chaos game and for it's successful creation it needs to follow one condition: the difference between the next vertex and the one previously chosen can't be one.\n" +

"To create it you need to do the following:\n\t" +

"1) Pick one of the pentagon vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point'\n " +

"The vertices' values are:\n " +

" 1 \n" +

"2 5\n" +

" 3 4\n";

### GIAPPONESE

ris = "この自己相似図形は混乱遊戯丈作られることができます。\n" +

"この自己相似図形の使い方は：\n\t" +

"一つ) 五角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点の値はーことができません。\n" +

"値は:\n" +

"\t一\t\n二\t\t五\n 三 四\n";

## Hive

### INGLESE

ris = "This fractal can be obtained only via chaos game and for it's successful creation it needs to follow one condition: the next vertex can't be the previous one.\n" +

"To create it you need to do the following:\n\t" +

"1) Pick one of the pentagon vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point'\n ";

### GIAPPONESE

ris = "この自己相似図形は混乱遊戯丈作られることができます。\n" +

"使い方は：\n\t" +

"一つ) 五角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点は同じ点ことができません。\n";

## Star

### INGLESE

ris = "This fractal can be obtained only via chaos game and for it's successful creation it needs to follow one condition: the next vertex can't be adjacent of one of the two vertices chosen previously.\n" +

"To create it you need to do the following:\n\t" +

"1) Pick one of the pentagon vertex as the starting point;\n\t" +

"2) Randomly choose one vertex and the starting point and put a midpoint between them which will become the next 'starting point\n ";

### GIAPPONESE

ris = "この自己相似図形は混乱遊戯丈作られることができます。\n" +

"使い方は：\n\t" +

"一つ) 五角形の頭頂は起点；\n\t" +

"二つ)次の点は先の点と次の点の半ばにあります。（次の点はランダムな起点ですよ）\n " +

"制約) 次の点は隣接点ことができません。\n";

## Felce di Barnsley

### INGLESE

ris = "Barnsley fern is a quite peculiar fractal.\n" +

"It is obtainable from a set of algorithms, each having its own percentage of being chosen:\n" +

"1%:\n\t x = 0\n\t y = 0.16 \* y\n" +

"7%:\n\t x = 0.2 \* x - 0.26 \* y\n\t y = 0.23 \* x + 0.22 \* y + 1.6\n " +

"7%\n\t x = -0.15 \* x + 0.28 \* y\n\t y = 0.26 \* x + 0.24 \* y + 0.44\n" +

"85%:\n\t x = 0.85 \* x + 0.04 \* y\n\t y = -0.04 \* x + 0.85 \* y + 1.6\n" +

"To change its shape (size, stretched, etc.) the coefficients' value has to be altered.\n" +

"x's range varies from 0 to 2.83, y's range varies from 0 to 9.999,\n";

### GIAPPONESE

ris = "バルネイーのフェーンは独特な自己相似図形です。\n" +

"使い方は：\n\t" +

"蓋然性 xの式 yの式 \n" +

"1%: x = 0 y = 0.16 \* y\n" +

"7%: x = 0.2 \* x - 0.26 \* y y = 0.23 \* x + 0.22 \* y + 1.6\n " +

"7%: x = -0.15 \* x + 0.28 \* y y = 0.26 \* x + 0.24 \* y + 0.44\n" +

"85%: x = 0.85 \* x + 0.04 \* y y = -0.04 \* x + 0.85 \* y + 1.6\n" +

##### "xの範囲は \n" + //TODO

##### "xの範囲は \n" + //TODO

## Set di Mandelbrot

### INGLESE

ris = "Mandelbrot's set is a particular fractal.\n" +

"For it's realisation the Argand plane (aka. Complex plane) must be used.\n" +

"It can be created using the following function:\n" +

"\tz(0) = 0\n\tz(n+1) = [z(n)]^(2)+c\n" +

"The black dots are the ones that stop growing after n iterations,\n" +

"The other dots have increasing color intensity equal to their function's rapid growth.\n" +

"But the particular thing is what happens at the edges." +

##### " "; //TODO but how?

### GIAPPONESE

ris = "マンダブロットのセットは奇妙な自己相似図形です。創造は優しいですから。\n" +

"　複素平面に作らなければなりません：\n\t" +

##### "xの式は " + //TODO

##### "" + //TODO IF THEN COLORI

"。\n";